

Biometric Attendance system Using R-305 Sensor and Arduino Uno

A Thesis report submitted in partial fulfillment of the requirements

for the award of the degree in

Master of Technology (Dual Degree)

in

“VLSI Design and Embedded Systems”

By

Sahil Suhag

(Roll No. : 711EC2128)

May, 2016

Under the guidance of

Prof. A.K Swain



Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela-769008



DEPARTMENT OF ELECTRONICS AND
COMMUNICATION ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY, ROURKELA
ODISHA, INDIA -769008

CERTIFICATE

This is to certify that the thesis entitled “**Biometric Attendance system**”, submitted by **SahilSuhag (Roll No. 711EC2128)** in partial fulfillment of the requirements for the award of **Master of Technology (Dual Degree) in VLSI Design and Embedded Systems** during session 2015-2016 at National Institute of Technology, Rourkela.

The candidate has fulfilled all the prescribed requirements. The Thesis is an authentically made, based on the candidates’ own effort and the implementation of his work.

To my knowledge and understanding, this thesis is up to the standard required for the award of a Bachelor of Technology Degree in Electronics and Communication Engineering and Master of Technology (Dual Degree) in VLSI Design and Embedded Systems.

Place: Rourkela

Prof. A.K Swain

Assistant Professor

Department of Electronics and Communication Engineering

National Institute of Technology

Rourkela -769008

ACKNOWLEDGEMENTS

I express my profound gratitude and indebtedness to Prof. A.K.Swain for giving me the present project topic and for his intellectual guidance, constructive criticism, inspiring me to further the scope of the project and valuable suggestions throughout the project work. He granted me relative freedom in deciding the course of action, providing valuable assistance wherever required.

I am also grateful to Kamal Kumar (Dept. of Computer Science and Engg.) for his help in the programming aspect of the project. All thought out the project his help in optimizing the code and error solving was very valuable.

I would also like to show my gratitude to Ratnakar Vijay (Dept. of Electronics and Comm. Engg.) for his help in the construction of the hardware.

I'm also thankful to the Dept. of Electronics and Communication Engineering NIT, Rourkela for their material support and making the facilities available for the project work.

Sahil Suhag

Roll No. 711EC2128

Contents

ABSTRACT.....	1
1 INTRODUCTION	5
1.1 Problem Statement.....	5
1.2 Motivation.....	5
1.3 Use of Biometrics	5
1.4 What is finger print?	6
1.5 Why use finger prints?	7
1.6 Attendance management system using finger print recognition	7
1.7 Organization Of Thesis	8
2 ARDUINO UNO	10
2.1 The Arduino Company and their Product	10
2.1.1 Software	10
2.2 The UNO Board	11
2.2.3 Communication.....	12
2.2.1 MEMORY	13
2.2.2 PINS	13
2.2.4 The Fail Safe:	14
2.3 ATMEGA328P PROCESSOR.....	15
3R-305 FINGERPRINT SCAN MODULE	18
3.1R-305	18
3.2 Buffer	19
3.3 Hardware connection.....	19
3.4 Communication Protocol	20
4. INTERFACING.....	23
4.1 Interfacing LCD with Arduino Uno	23
4.2 Interfacing R-305 with Arduino Uno	24
4.3 Interfacing the command buttons	24
4.4 Connection With Arduino	24
5 BIOMETRIC SYSTEM SOFTWARE ARCHITECTURE.....	27
5.1 Flowchart	27

5.2 Subroutines	28
5.2.1 SCAN PRINT	29
5.2.2 STORE PRINT	29
5.2.3 LOAD PRINT	30
5.2.4 MATCH PRINT.....	30
5.3 Routines	31
5.3.1 ADD FINGERPRINT.....	31
5.3.2 DELETE FINGERPRINT	32
5.2.3 SCAN FINGERPRINT	32
6 CONCLUSION AND FUTURE SCOPE OF THE PROJECT.....	34
6.1 Conclusion	34
6.2 Future Scope of the Project	34
REFERENCES	36

ABSTRACT

Every organization, be an educational institute or organization public or private, it has to maintain accurate records of attendance of students or staff for effective working of organization. As time has shown, both private and public sectors face high level of mismanagement due to false records and impersonations. Employers and officials are concerned over employee absenteeism in their human resources and the problems in maintaining records of student attendance during lecture periods. Finger prints are the patterns of ridges and valleys on everyone's finger tips. They can be used as a type of biometric identification which, like all personal features, are unique to a person and do not change in the person's lifetime. An attendance system based on fingerprint technology, suitable to be used in a university environment, is presented in this thesis. The finger print based attendance system was implemented with Arduino Uno Microcontroller, R-305 fingerprint sensor and programmed in C++ using Arduino IDE. It comprises of two processes; enrollment and authentication. During enrollment, the finger print of the user is captured and its unique defining features extracted to be stored in the flash memory along with the other users identities as a template. Minutiae points, the unique feature, are extracted using a method which extracts the ridge's endings and bifurcation. During authentication, the user's fingerprint is scanned to be stored in one of the two buffers and the extracted features compared with the template which is loaded to the other buffer to compare the match before attendance is verified. The experimental result exhibits that the system developed is highly efficient in verifying fingerprint with a high level of accuracy.

LIST OF FIGURES

Figure No.	Figure Title	Page No.
Figure 1	Minutia Points	14
Figure 2	AnArduino Uno with its Different Parts Labeled.....	20
Figure 3	The ATmega328P Microcontroller	24
Figure 4	A R-305 Fingerprint Module.....	26
Figure 5	Pin Numbering of the two Interfaces.....	28
Figure 6	Block Diagram of LCD Pin Connection	31
Figure 7	Block Diagram Showing the Connection of Pins of the System.....	33
Figure 8	Flowchart Showing the Execution of the Program.....	35

LIST OF TABLES

Table No.	Table Title	Page No.
Table 1	Specifications of Arduino Uno	21
Table 2	Specification of R-305 Fingerprint Module.....	27
Table 3	Pins Names and Function	28
Table 4	Definition of Data Package.....	29

CHAPTER 1

INTRODUCTION

- Motivation
- Organization of thesis

1 INTRODUCTION

1.1 Problem Statement

Designing and implementing a student attendance verification system based on finger print recognition and is capable to manages records for attendance in classrooms, laboratories seminars etc. in institute like NIT Rourkela.

1.2 Motivation

Maintaining a record for employee attendance has been in practice for a long time. Over the years the system to maintain these records has been made more efficient with the introduction of various technologies. Designing an improved attendance management system for students so that records can be maintained with ease, transparency and accuracy was an important motivating factor behind this project. The accuracy of attendance records would be improved as it will eliminate the hassles of roll calling, where a person physically observes and verifies attendance, and therefore will be crucial in saving time of both – the students and the teachers. Various image processing techniques to recognize fingerprint patterns are available. Today in terms of technology they are very advanced and hence very precise. Hence a fast and accurate attendance system, free from the possibility of impersonation is designed.

1.3 Use of Biometrics

Biometric Identification based systems are in wide use for the purpose of uniquely singling

out human beings basically to identify and verify them. There are a lot of biometric based systems like facial feature recognition, voice recognition, eye's iris recognition, fingerprint recognition, palm recognition etc. In this project, we have utilized a finger print recognition system. Such biometric systems can also find usage to provide identity access control. Hence the use of biometric fingerprint recognition in a student attendance management system is a well trodden path and a reliable approach.

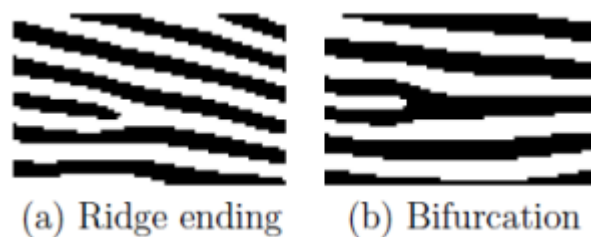


Figure 1. Minutia Points

1.4 What is finger print?

The intricate pattern of ridges and valleys on the surface of a fingertip is called a fingerprint. The ridges are described as the portion of the print which protrudes out and valley is the sunken area between them. The minutiae points are where the ridges end or are split. The minutiae pattern of each finger, like all other personal features, are unique to a person and do not change in the person's lifetime. When a ridge curve terminates or ends this is defined as a ridge ending. Whereas when a ridge separates from a single path into two forming a Y-junction, the split is defined as a bifurcation. Figure 1 displays a ridge ending and a bifurcation. In this example, the dark coloured portion corresponds to the ridges, and the pale portion corresponds to the valleys. When a machine or a human fingerprint expert determines whether two given fingerprints belong to the same finger, the degree of match between two minutiae pattern is one of the most detrimental factors. Thanks to the similarity in the way a human finger print expert determines unique matches and compactness of the

size templates, the minutiae-based fingerprint matching technique is the most widely studied matching method.

1.5 Why use finger prints?

At the present level of development fingerprints have been proven to be the easiest and fastest method for biometric identification. They are secure to use, unique for every person and do not change in a person's lifetime. Besides these, utilization of fingerprint recognition system is relatively cheap, easy to understand and to put into practice and is accurate to satisfy the purpose. Fingerprint based recognition systems have been put to wide use in both forensic and civilian applications. In comparison with other available biometrics feature matching systems, fingerprint-based biometrics is the most developed technique and holds the largest market shares. As these systems have been optimized over the years, not only is it faster than other techniques but the amount of energy consumed by such systems is also less.

1.6 Attendance management system using finger print recognition

Maintaining attendance records of students of an institution as large as ours is a tedious task. It takes both time and paper. To change all work regarding attendance to an automated system and on-line, I have built an attendance recording system which can be installed in an institute. This project uses a finger print scanning module(R-305) interfaced with a microcontroller (Arduino Uno). This fingerprint identification system uses existing techniques in fingerprint recognition and matching.

1.7 Organization of Thesis

Chapter 1: This section has an introduction and the organization of thesis

Chapter 2: This chapter gives information regarding Arduino Uno microcontroller. The memory of the device, input/ output pins and other hardware information is described. The functioning of ATmega328P microcontroller used in Arduino Uno is also given.

Chapter 3: This chapter deals with the R-305 fingerprint module. All specifications about the device are mentioned. Its communication protocol is also described.

Chapter 4: This chapter describes the interfacing of various devices to the Arduino Uno, namely the LCD, the R-305 module and the four command buttons.

Chapter 5: This chapter explains the working of the software that runs our system. How various routines and subroutines are executed is also described.

Chapter 6: This chapter gives the conclusion and the scope of future developments with regard to this project.

Chapter 7: This chapter states the references utilized to accomplish this project work.

CHAPTER 2

ARDUINO UNO

- Arduino Uno
- ATmega328PProcessor

2 ARDUINO UNO

2.1 The Arduino Company and their Product

Arduino is an Italy based software and hardware company and user community that designs and manufactures computer open-source hardware and software. It basically produces microcontroller-based kits for the purpose of building practical digital devices and interactive projects that are capable of sensing and controlling physical devices. First introduced in 2005, the Arduino aimed to provide a low cost, easy way for amateurs and skilled technicians to create devices that interact with their environment and surroundings using sensors and actuators. Commercially available Arduino MCU Boards come in preassembled form, or as do-it-without anyone's help kits. The hardware design plans too are freely available, allowing the Arduino MCU boards to be made by any and everyone. In 2013, Adafruit Industries assessed that around 700,000 official boards were in its client's hands.

2.1.1 Software

The Arduino company provides the Arduino integrated development environment (IDE), this is a multiple-platform application developed in Java. To let electronics enthusiasts get familiarized with software development, its designed to introduce them to programming. It contains a code editor having features such as syntax highlighting, brace matching, and automatic indentation, and provides quick and simple one-click operations to compile and then upload programs on to an Arduino MCU board. A program written on the ArduinoIDE

is known as a "sketch".

With a few special rules to organize the code, the Arduino IDE basically uses C and C++ languages. The Arduino IDE comes equipped with a software library, which has many commonly used I/P and O/P procedures. A typical Arduino C/C++ program would comprise of two functions that are compiled and linked together forming the program. The sketch is a cyclic executive program:

setup(): this function runs once at the beginning of a sketch and is used to initialize devices/settings.

loop(): a function called repeatedly till the device is in use. This runs until the Arduino board is powered off.

After the GNU toolchain has been linked and the program compiled, the Arduino IDE uses a program to convert the executable code or the program into a text file in hexadecimal coding. Using a loader software provided in the firmware of the board, the program is then loaded into the Arduino Uno board.

2.2 The UNO Board

The Uno is a MCU board based on the ATmega328P. It has been provided with 14 digital I/O pins (of these 6 can be employed as PWM O/Ps), 6 analog I/Ps, a USB connection, a 16 MegaHz quartz crystal, an ICSP header, a power jack and a reset button. It has everything required to support the microcontroller; we just got to connect it to a computer or any such processing device with the provided USB cable or it can be powered using a 5V- 12V AC-to-DC adapter or appropriate battery to begin operation. We can experiment with our Uno without thinking about doing something wrong, at worst we may damage the MCU. This can

be easily rectified by replacing the chip cheaply and just as easily starting over again.

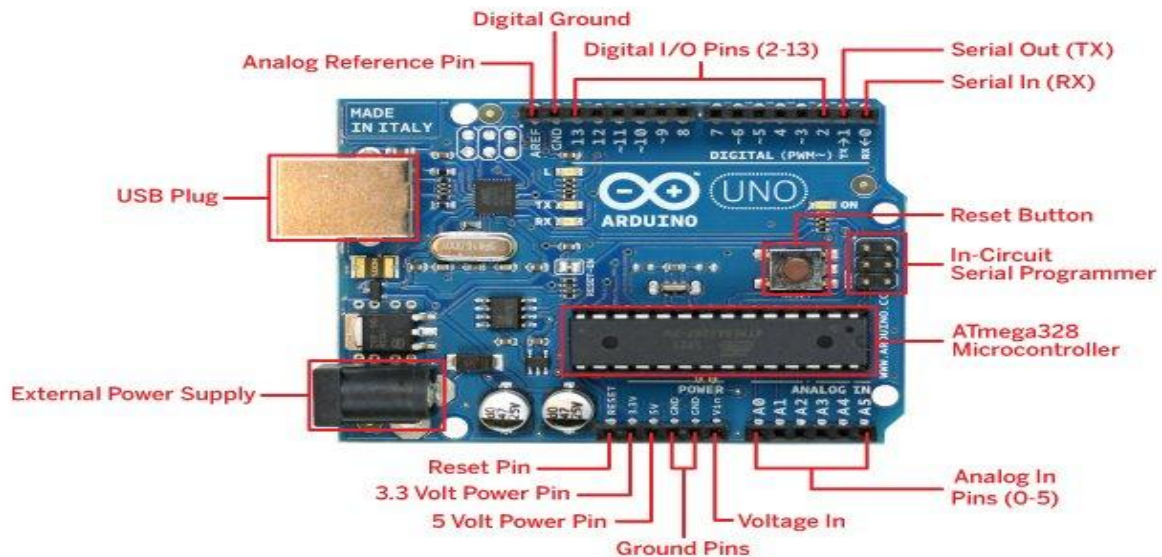


Figure2. An Arduino Uno with its Different Parts Labeled

2.2.3 Communication

The Uno contains a number of facilities and methods to communicate with the upper computer, other Uno boards, or with other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channelizes this serial communication on the USB and it will appear as a virtual com port to Arduino IDE on the upper computer. No external driver is needed as the 16U2 firmware uses the standard USB COM drivers. However, for Windows operating system, a .inf file is needed. The Arduino Software (IDE) also has a serial monitor, which serves as a output, this allows simple textual data to be sent and

received from the board.

Serial communication on any of the Uno's digital pins is made possible using the SoftwareSerial library provided with the Arduino IDE.

ATmega328P also supports I2C (TWI) and SPI Communication. To simplify use of the I2C bus the Arduino Software (IDE) comes with a Wire library for the purpose.

2.2.1 MEMORY

The ATmega328 comes with 32 KiloByte (bootloader utilizes 0.5 KB of the flash). It also contains 2 KiloByte of Static RAM and also 1 KiloByte of EEPROM (on which read and write is performed with the help EEPROM library).

2.2.2 PINS

Arduino Uno has been provided with 14 Digital Pins which can be used as inputs or outputs on user discretion. Using the function `pinMode()`, `digitalRead()` and `digitalWrite()` the pins can be initiated. 5 volts is standard operational voltage for each of the pins. A current of 20 mA flows through these pins under the recommended operating. A maximum current value of 40mA mustn't be exceeded on any Input/Output pin to avoid causing irreparable damage to the microcontroller.

Some pins have specific purpose:

Serial Pins: 0 (RX) I/P and 1 (TX) O/P. The use of these two pins is to receive (RX) and transmit (TX) TTL serial data. The RX and TX LEDs blink when data is transmitted on these pins.

PWM: Pin numbers :3, 5, 6, 9, 10, and 11. Using the `analogWrite()` function these pins provide 8-bit PWM O/P .

LED: 13. The digital pin 13 drives a built in LED. If the pin has HIGH value, the LED glows, when the pin has a LOW value, it's off.

Six clearly labeled pins A0 - A5 are the Arduino Uno's 6 analog input pins, each providing 10 bits of resolution (that is 1024 different digital values). By default the upper end of their range is 5 volts and the lower end of the range is at volts. This can be altered using the AREF pin and the `analogReference()` function.

There are a few more other pins on the board:

Reset: To reset the microcontroller this line is set to LOW by pressing the reset button.

2.2.4 The Fail Safe:

To protect our computer's internal parts and its USB ports from shorts and overcurrent, the Uno contains a resettable polyfuse. It is known that most computers come with their own system of internal protection, the fuse acts as an extra layer of safety. If a current of value more than 500 mA is drawn to or from the USB port, the fuse would automatically break the circuit till the short or overload is rectified.

TABLE 1. Specifications of Arduino Uno

Microcontroller	ATmega328P
Operating Voltage	5V
Input Voltage (recommended)	7-12V
Input Voltage (limit)	6-20V
Digital I/O Pins	14 (of which 6 provide PWM output)
PWM Digital I/O Pins	6
Analog Input Pins	6
DC Current per I/O Pin	20 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328P) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Clock Speed	16 MHz
Length	68.6 mm
Width	53.4 mm
Weight	25 g

2.3 ATMEGA328P PROCESSOR

The ATmega328P is low power Complementary MoS 8bit microcontroller, the architecture of this is based up on the AVR enhancement Reduced instruction set computer architecture. It will execute the instruction in a single clock cycle even the instructions are powerful. ATmega328P can achieve high throughputs approximately 1 MIPS per MHz, so the designer has the full flexibility to design and optimize between power consumption and processing speed.

CHAPTER 3

R-305 FINGERPRINT SCAN MODULE

3R-305 FINGERPRINT SCAN MODULE

3.1R-305

This is a finger print sensor module using a TTL UART interface. The fingerprint information saved inside the module can be stored by the user and can be configured in either 1:1 or 1: N mode for identifying the person. A 3v3 or 5v Microcontroller can be directly interfaced with R-305. When enrolling, user is required to scan finger twice. The system will independently process the two finger image scans, based on processing results generate a template of the finger and then store it. During match routine, user scans the finger tip via the optical sensor. System generates a template of the fingerprint and compares it with the templates stored in the finger library, stored in the flash. For 1:1 matching, system will compare the live fingerprint with templates stored in the Module; for 1:N matching, the system will go through the whole finger library and servers if any to find the matching user. In both circumstances, the matching result, success or failure will be returned by the system.

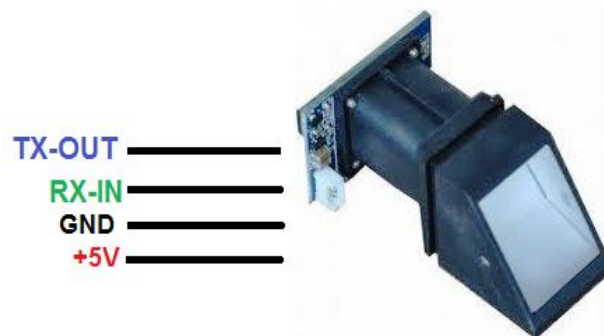


Figure4.A R-305 Fingerprint Module

TABLE 2. Specification of R-305 Fingerprint Module

• Fingerprint sensor type: Optical
• Sensor Life: 100 million times
• Static indicators: 15KVBacklight: bright green
• Interface: USB1.1/UART(TTL logical level)
• RS232 communication baud rate: 4800~115200bps (changeable)
• Dimension: 55*32*21.5mm
• Image Capture Surface 15—18(mm)
• Verification Speed: 0.3 sec
• Scanning Speed: 0.5 sec
• Character file size: 256 bytes
• Template size: 512 bytes
• Storage capacity: 250
• False Acceptance Rate (FAR) :0.0001%
• False Rejection Rate (FRR): 0.1%
• Resolution 500 DPI
• Voltage :3.6-6.0 VDC
• Working current: Typical 90 mA, Peak 150mA
• Matching Method: 1: N
• Operating Environment Temperature: -20 to 45° centigrade.

3.2 Buffer

The module contains an image buffer and two512-byte-character-file buffer within the module's RAM space. The buffers can read & written by the user via instructions. Note: Data/ Contents of the mentioned buffers will be lost once the power is off.

3.3 Hardware connection

The MCU of 3.3V of 5V power and the Module may communicate via serial interface: TD connects with RX (MCU receive pin), RD connects with TX (MCU transfer pin).

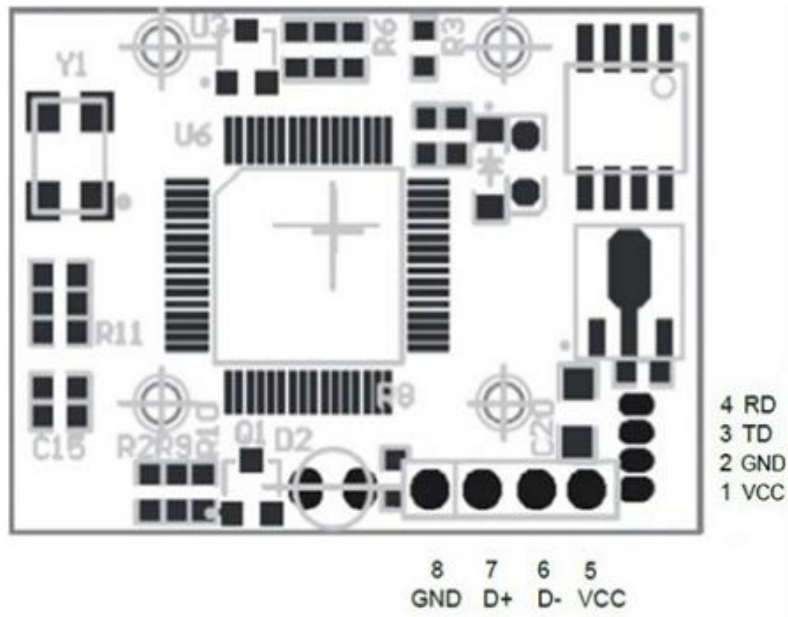


Figure5. Pin Numbering of the two Interfaces

TABLE 3. Pins Names and Function

Pin No.	Name	Function
1.	VCC	Power input
2.	GND	Signal ground
3.	RD	Data input. TTL logic
4.	TD	Data output. TTL logic
5.	D+	Data +
6.	D-	Data-
7.	VCC	+5 VDC
8.	GND	Ground

3.4 Communication Protocol

When communicating, the transfer and receiving of command/data or result is done by wrapping all of them in data package format.

Data package format

Header	Adder	Package identifier	Package length	Package content (instuction/data/Parameter)	Checksum
--------	-------	--------------------	----------------	---	----------

TABLE 4. Definition of Data Package

Definition of Data package

Name	Symbol	Length	Description	
Header	Start	2 bytes	Fixed value of 0xEF01; High byte transferred first.	
Adder	ADDER	4 bytes	Default value is 0xFFFFFFFF, which can be modified by command. High byte transferred first and at wrong adder value, module will reject to transfer.	
Package identifier	PID	1 byte	01H	Command packet;
			02H	Data packet; Data packet shall not appear alone in executing processs, must follow command packet or acknowledge packet.
			07H	Acknowledge packet;
			08H	End of Data packet.
Package length	LENGTH	2 bytes	Refers to the length of package content (command packets and data packets) plus the length of Checksum(2 bytes). Unit is byte. Max length is 256 bytes. And high byte is transferred first.	
Package contents	DATA	—	It can be commands, data, command' s parameters, acknowledge result, etc. (fingerprint character value, template are all deemed as data);	
Checksum	SUM	2 bytes	The arithmetic sum of package identifier, package length and all package contens. Overflowing bits are omitted. high byte is transferred first.	

CHAPTER 4

INTERFACING WITH ARDUINO

- Interfacing LCD
- Interfacing R-305
- Interfacing Command Buttons

4. INTERFACING

4.1 Interfacing LCD with Arduino Uno

Data pins D4 to D7 are connected to digital pin no. 5, 6, 7 and 8 of the Arduino.

Pin No. 1, 4 (R/W) and pin 16 are grounded.

Pin No. 2 and 15 are at +5V.

RS (Register select) is connected to pin 3 of Arduino.

E (Enable) is connected to pin 4 of Arduino.

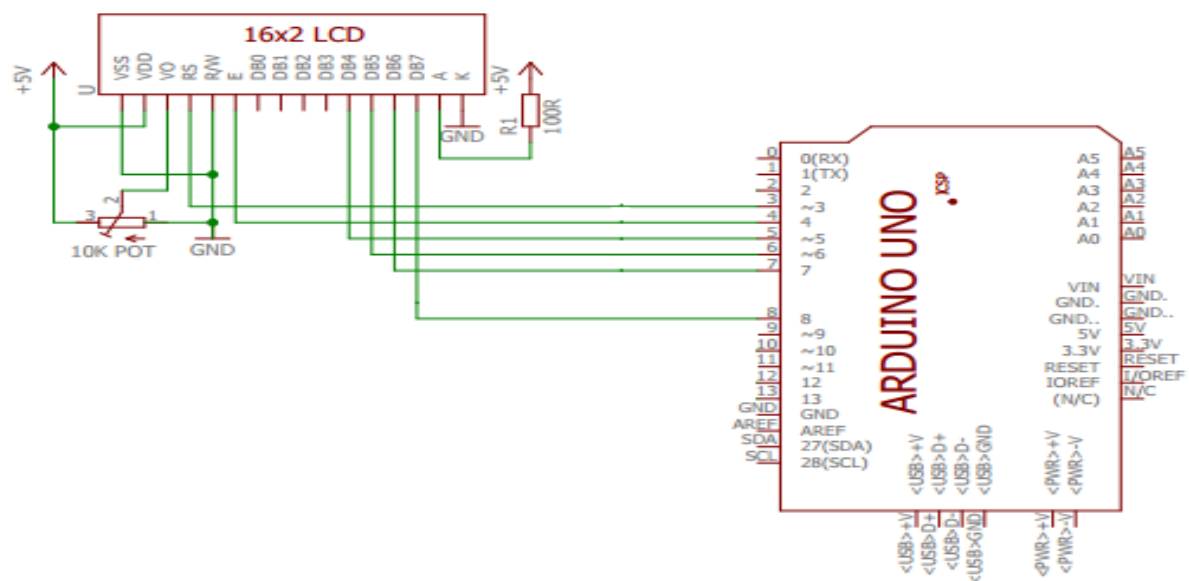


Figure6. Block Diagram of LCD Pin Connection

The LCD pins are connected to the digital and power pins of the Arduino. The LCD library available in the Arduino IDE is included in the code. Pins 3, 4, 5, 6, 7, 8 are initiated.

4.2 Interfacing R-305 with Arduino Uno

The module is powered up with +5 V from the Arduino and the GND.

RX is connected to Digital pin 10

TX is connected to Digital pin 11; serving as input and output respectively.

4.3 Interfacing the command buttons

Four buttons are designated for ADD, VERIFY, DELETE and SELECT. They trigger the routines which carry these functions. The buttons are connected to the pins as:

ADD - SW(1) is connected to pin 2

DELETE - SW(2) is connected to pin 9

VERIFY - SW(1) is connected to pin 12

SELECT - SW(4) is connected to pin 13

4.4 Connection with Arduino

As described above the connection of various pins are soldered together. A voltage source regulator is used to supply 5V supply to the system. This is in case there is no electricity. Further the Arduino Uno and the R-305 module were pasted on the PCB using plastic adhesive. The PCB is in turn stuck to a card board. To improve accessibility the command buttons are pasted on a small patch of PCB and stuck further away from the main system.

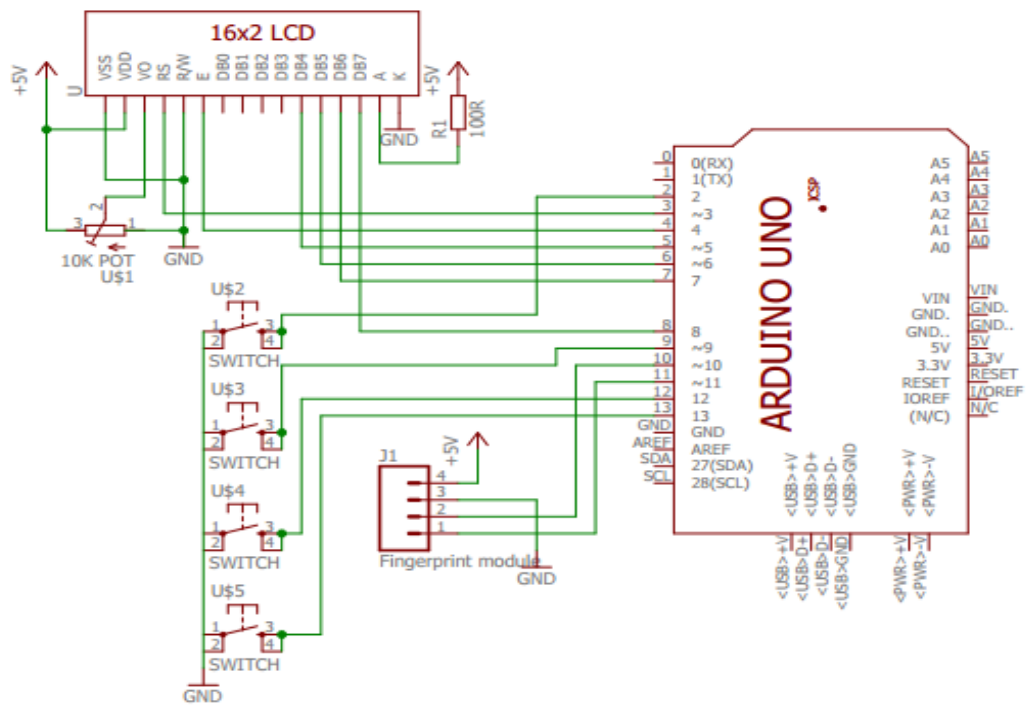


Figure7. Block Diagram Showing the Connection of Pins of the System

CHAPTER 5

BIOMETRIC SYSTEM SOFTWARE ARCHITECTURE

- Flowchart
- Subroutines
- Routines

5 BIOMETRIC SYSTEM SOFTWARE ARCHITECTURE

5.1 Flowchart

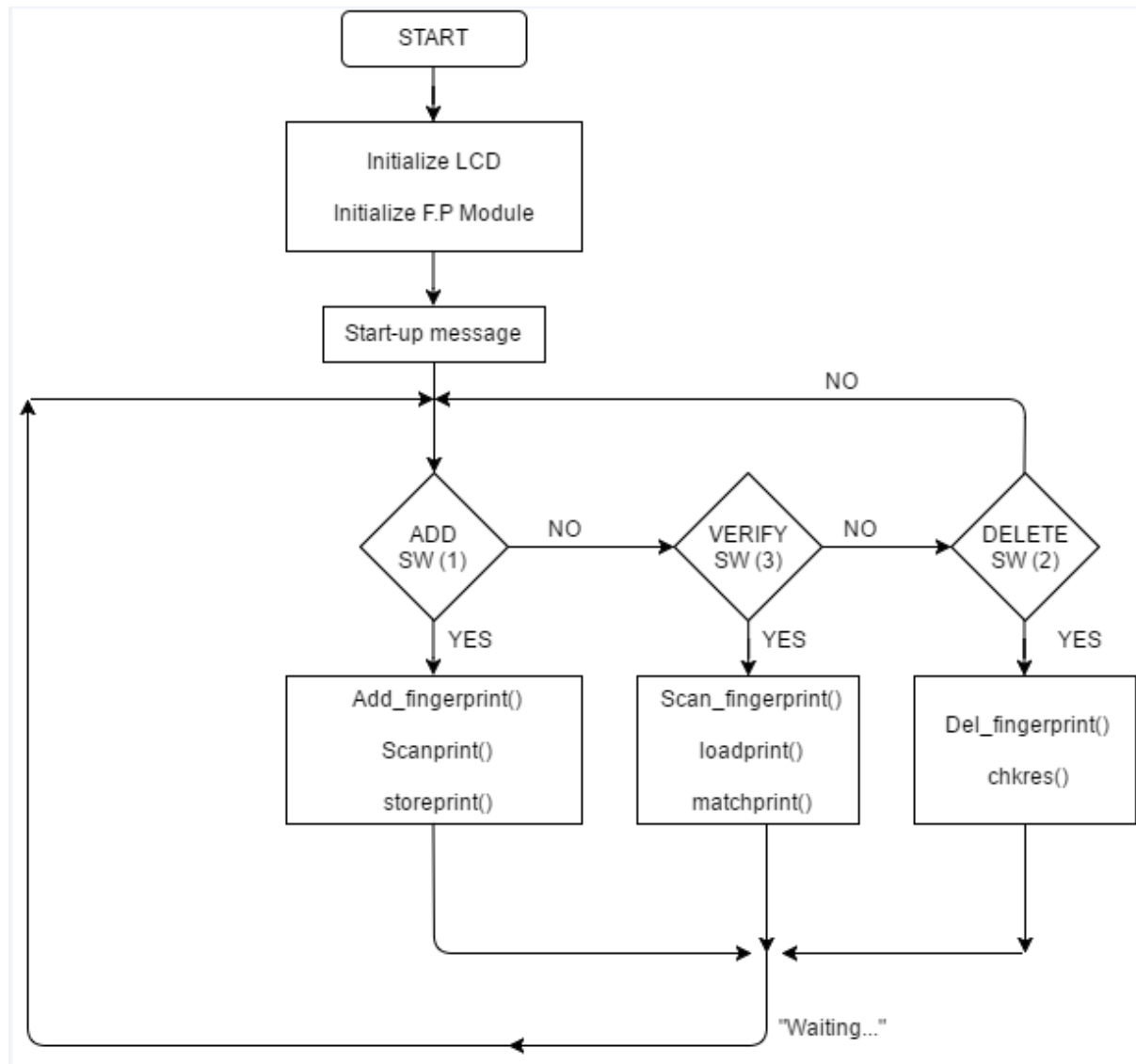


Figure8.Flowchart Showing the Execution of the Program

Upon the execution of this program the various processes will begin functioning. These processes systematically follow the instructions in the flowchart. Upon powering up the system the machine stays in “waiting” mode, this is the idle state in which the machine stay

until a command is given. On the execution of the command and the routines that follow the system again goes into the idle state, until the next command is issued.

So upon starting the module the R-305 sensor module and the LCD screen are initiated i.e. they handshake with the Arduino board. If this is successfully achieved the LCD screen lights up and displays the message “Fingerprint System” followed by the message “Fingerprint OK”. The first message is the welcome message and the next message confirms the successful handshake between the fingerprint module and the Arduino Uno. Now the system stays in idle mode waiting for a command instruction – ADD, DELETE or VERIFY.

Upon pressing any of the above three command buttons the corresponding routine is executed. The program identifies the key pressed. This initiates the routine to be carried out next. The system will perform the function of Enroll, Match and Delete. Upon any of these functions being called the program will run the following subroutine. In *add_fingerprint()* the user’s fingerprint is scanned and stored in one of the five memory location in the flash. In *scan_fingerprint()* the user’s finger is scanned and stored in one of the two buffer and the print is verified against all the saved users. Upon finding a successful match the user I.D. is displayed on the LCD, else a match not found message is displayed. In *del_fingerprint()* the desired user I.D. is selected and the corresponding memory is emptied. All memory location selections are made using the fourth “SELECT” button. The detailed working of the routines and subroutines are as follows:

5.2 Subroutines

The process of storing an image in template form in the flash memory is as follows:

1. Take Image
2. Generate Character file

3. Generate Template from Character file
4. Store template to Flash Memory

5.2.1 SCAN PRINT

This subroutine first takes two fingerprint image scan one after another. These two scans, stored in both the buffers, are superimposed and the minutia points extracted. The process of storing first requires an image to be scanned. This scanned image is converted into a character file and stored in one of the two buffers. Another scan is taken and compared with the first to make sure the minutia points of the two scans match. This generated character file of the fingerprint scan is ready to be converted into a template to be further stored in the flash memory.

Pseudo Code:

1. Send instruction to module to scan fingerprint
2. Scan fingerprint image.
if (failure) scan problem;
3. Generate char file and store in buffer 1
4. Take another scan and char file image stored in buffer 2
if(successful) scan acceptable;
else scan failed;

5.2.2 STORE PRINT

This subroutine first converts the character File into a template. The template is then stored in

a designated location of the Flash memory. Here the locations are numbered is 1 to 5. These numbers represent the page number in the flash memory of the R-305 module where the templates get stored. Each page stores one template of size 512 bytes. The locations of these pages are stored in the EEPROM of the Arduino Uno.

Pseudo Code:

1. Send instruction to module to convert the character file into template
2. Store template in one of the five locations of the memory chosen
3. Update value of EEPROM.

5.2.3 LOAD PRINT

This instruction is used to load templates at a specified location (Page I.D.) of Flash library to buffer CharBuffer1/ CharBuffer2. Here the page number (which is called) is added to buffer I.D. and page I.D. The template is loaded in one of the two buffers to be compared or matched.

Pseudo Code:

1. Call Page Number
2. Template loaded into the buffer
if (load successful); display user I.D.
else load fail;

5.2.4 MATCH PRINT

This instruction is used to carry out exact matching of templates from both the buffers -

CharBuffer1 and CharBuffer2. The scanned image is stored in one of the two buffers. All the images are sequentially loaded into the other buffer and compared. Upon finding the correct match the user I.D. is displayed on the LCD. Else a match not found message is displayed.

Pseudo Code:

1. Send instruction to module to scan fingerprint and load buffer 1
2. Call all stored templates sequentially into buffer 2
3. Match each template with buffer 1
4. if (match found); display user I.D.
 else match not found;

5.3 Routines

5.3.1 ADD FINGERPRINT

This routine is called upon pressing the ADD switch. First “user-1” appears on the LCD screen. Again pressing the ADD button the next “user-2” is displayed. If “ok” is displayed alongside the user I.D. mean yes, that memory slot is occupied, whereas if “X” appears that means the memory location is empty and a template can be stored. If the select button is pressed the memory location will store the fingerprint against that user.

Pseudo Code:

1. Display users with I.D.
2. Select a user I.D.
3. Scan print
4. Store print

5. Update EEPROM values

5.3.2 DELETE FINGERPRINT

This routine is called upon pressing the delete switch. This function operates in the same manner as the Add Fingerprint routine. On selection of a user the print against it is deleted. The memory location is emptied and appropriate message to denote this is displayed.

Pseudo Code:

1. Display users with I.D.
2. Select a user I.D.
3. Template deleted

5.2.3 SCAN FINGERPRINT

On pressing the verify button this routine gets initiated. This function first asks the person to scan his/her finger. This print is stored in one of the two buffers. Then sequentially all the stored prints are loaded to the other buffer and compared. On finding a match the LCD displays the user I.D. Upon failing to find a match, a “match not found” message is displayed.

Pseudo Code:

1. Input fingerprint
2. Store in buffer 1
3. Call all user sequentially in buffer 2
4. Match prints
5. Display result

CHAPTER 6

CONCLUSION AND FUTURE PROSPECTS FOR THE PROJECT

6 CONCLUSION AND FUTURE SCOPE OF THE PROJECT

6.1 Conclusion

Hence a biometric attendance system was successfully designed and implemented. Using the four buttons one can perform the desired operation upon the system. Developing cheap and easy to operate systems like these I hope we can effectively monitor the attendance of employees, students etc. effectively and without the hassles of roll calling, proxies & human-error. This will save valuable time and effort, also virtually removing the possibility of human error. Such a system is fully automated, with records which can accessed easily being generated. This would increase transparency and decrease the management required for the proper maintenance of these records.

6.2 Future Scope of the Project

This system could be connected to a server and daily records stored and maintained. The flash of the R-305 module can store up to 250 image scans, implying that such a system can easily accommodate the needs of a classroom. Given the strict standards of attendance at NIT, Rourkela; a biometric attendance system is the way forward to maintain accurate records. Furthermore as it an individual identification system it can be employed as door lock to grant access to a facility like a home, an office etc.

REFERENCES

REFERENCES

- [1] **Biometric Systems** - Technology, Design and Performance Evaluation By - James Wayman, AnilJain, Davide Maltoni and Dario Maio (Eds). Springer Publications.
- [2] On Board Manual for organization to install Aadhar – enabled biometric attendance system. Issued from : National Informatics Centre (NIC). Govt. Of India. Nov, 2014.
- [3] Open source site for code help, www.Circuitoday.com
- [4] Connection assistance from www.arduino.com
- [5] en.wikipedia.org/wiki/Arduino
- [6] FINGERPRINT BASED LICENSE CHECKING FOR AUTO-MOBILES. IEEE Conference Paper. By - J.Angeline Rubella, M.Suganya B.Santhosh Kumar, K.R.Gowdham, M.Ranjithkumar, K.Senathipathi.
- [7] Open source code assistance from www.stackexchange.com
- [8] A Robust Embedded Biometric Authentication System Based on Fingerprint and Chaotic Encryption. Elsevier Journal. By-M.A. Murillo-Escobar and C. Cruz-Hernández.